

## AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings of claims in the application:

### LISTING OF CLAIMS

1-58 (Cancelled)

59. (Currently Amended) A small footprint device comprising:

at least one processing element configured to execute groups of one or more program

modules in separate contexts, said one or more program modules comprising

zero or more sets of executable instructions and zero or more sets of data

definitions, said zero or more sets of executable instructions and said zero or

more data definitions grouped as object definitions, each context comprising a

protected object instance space such that at least one of said object definitions is

instantiated in association with a particular context;

a memory comprising instances of objects; and

a context barrier for separating and isolating said contexts, said context barrier

configured for controlling execution of at least one instruction of one of said zero

or more sets of instructions comprised by a program module based at least in part

on whether said at least one instruction is executed for an object instance

associated with a first one of said one or more separate contexts and whether said

at least one instruction is requesting access to an instance of an object definition

associated with a second one of said one or more separate contexts, said context

barrier further configured to prevent said access if said access is unauthorized and enable said access if said access is authorized, The small footprint device of claim 1 wherein an object instance is associated with a context by recording the name of said context in a header of said object instance, information in said header inaccessible to said one or more program modules.

60-61 (Cancelled)

62. (Currently Amended) A method of operating a small footprint device that includes a processing machine, wherein program modules are executed on the processing machine, the method comprising:  
executing groups of one or more program modules in separate contexts, said one or more program modules comprising zero or more sets of executable instructions and zero or more sets of data definitions, said zero or more sets of executable instructions and said zero or more data definitions grouped as object definitions, each context comprising a protected object instance space such that at least one of said object definitions is instantiated in association with a particular context;  
and  
providing a context barrier for separating and isolating said contexts and for controlling execution of at least one instruction of one of said zero or more sets of instructions comprised by a program module based at least in part on whether said at least one instruction is executed for an object instance associated with a first one of said one or more separate contexts and whether said at least one

instruction is requesting access to an instance of an object definition associated with a second one of said one or more separate contexts, said providing further comprising:

preventing said access if said access is unauthorized; and

enabling said access if said access is authorized, ~~The method of claim 37~~

wherein an object instance is associated with a context by recording the name of said context in a header of said object instance, information in said header inaccessible to said one or more program modules.

63-64. (Cancelled)